Available online at www.sciencedirect.com
ScienceDirect
Procedia Engineering 96 (2014) 489 – 499
Procedia Engineering
www.elsevier.com/locate/procedia

Modelling of Mechanical and Mechatronic Systems MMaMS 2014

Analyzing, Modeling and Simulation of Humanoid Robot Hand Motion

Ivan Virgala[a]*, Michal Kelemen[a], Martin Varga[b] , Piotr Kuryło[c]

[a]*Technical University of Košice, Faculty of Mechanical Engineering, Department of Mechatronics, Letná 9, 04200 Košice, Slovakia*
[b]*ZTS VVÚ KOŠICE, Južná Trieda 95, 041 24 Košice, Slovakia*
[c]*University of Zielona Góra, Licealna 9 st., 65-247 Zielona Góra, Poland*

Abstract

The paper deals with analyzing, modeling and simulation of motion of humanoid robot hand. At first, the human hand is introduced from the view of biology with focus on bones and joints. Particulars places of motion are replaced by corresponding mechanical joints. Subsequently the kinematic configuration of humanoid hand consisting of 24 degrees of freedom is designed. New method for inverse kinematic model is introduced using Matlab functions. Next, dynamic model of humanoid hand is introduced using model-based design by means of Matlab / SimMechanics. The result of the work is model in Matlab, which can control of particular fingers motion.

Peer-review under responsibility of organizing committee of the Modelling of Mechanical and Mechatronic Systems MMaMS 2014

Keywords: humanoid hand; mathematical model; matlab; simulation;

1. Introduction

Service robotics reaches to our everyday life in many fields even though people do not realize it. Recent developments in robotics show a growing interest in assistance and personal robots. These robots found their application in areas like industrial companies, military, health service (care of elderly people, nursing robots, etc.), or like household robots and guiding robots [1, 5]. Particular parts of human body of people who lost their limbs can be

* Ivan Virgala. Tel.: +421-905-723-153.
E-mail address: ivan.virgala@tuke.sk

Peer-review under responsibility of organizing committee of the Modelling of Mechanical and Mechatronic Systems MMaMS 2014
doi:10.1016/j.proeng.2014.12.121

replaced by robotic mechanism like arm, hand or leg. From this reason biomechanics is important field of research [7,8, 10]. Our study is focused on modeling of humanoid robot hand in order to design a real model which should be used in health service areas. The first important contributions in the area of humanoid robotics came in 70s and 80s [2]. Many researchers have focus on realization of humanoid robots. Some of them are focused on robot walking, some of them are focused on robot hand [3]. Our team would like to continue basing on previous work and knowledge. The aim of the paper is modeling and simulations of motion of humanoid robot hand.

2. Biomechanics of human hand

For suitable designing and modeling of humanoid robot hand it is necessary to study human hand from the view of biology. Particular joints and bones so can be replaced by revolute or universal joints and links are considered as rigid bodies. Human hand consists of 14 phalanges, 5 splint bones and 8 wrist bones, see Fig. 1 - 2.

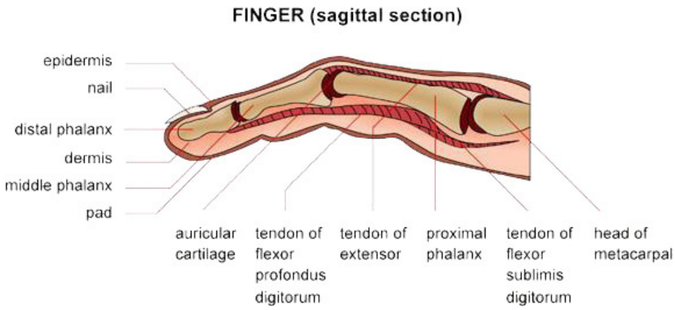


Fig. 1. Profile of a human hand finger

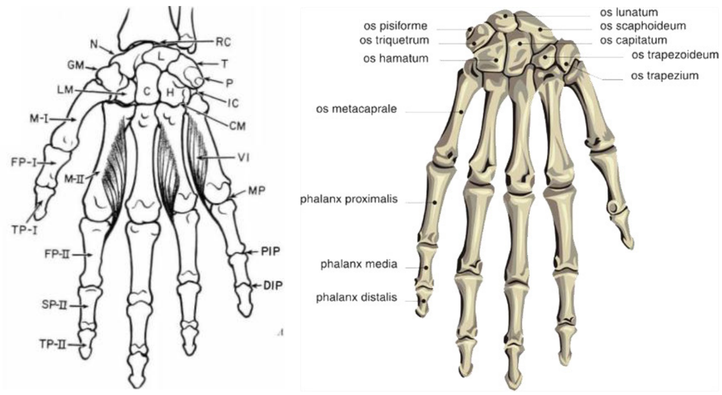
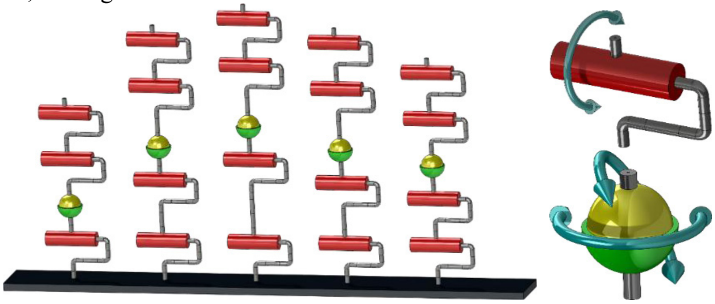


Fig. 2. Parts of human hand [4]

Tab. 1 Bones and joints in human hand

Bones:	**Joints**:
• Carpal - GM, N, L, T, P, LM, C, H	• Radiocarpal – RC
• Metacarpal - M I, II, III, IV V	• Intercarpal – IC
• First Phalangeal - FP I, II, III, IV, V	• Carpometacarpal – CM
• Second Phalangeal - SP II, III, IV, V	• Metacarpophalangeal – MP
• Third Phalangeal - TP I, II, III, IV, V	• Proximal Interphalangeal – PIP
	• Distal Interphalangeal – DIP

From the Fig. 2 can be seen the complexity of human hand. By carpal bones the hand is connected to forearm and it can be neglected for modeling of hand motion. Metacarpal bones form a palm of a hand by their accurate shape and their motion is limited. Carpometalcarpal, proximal interphalangeal and distal interphalangeal joints can be considered as revolute joints and metacarpophalangeal joint as ball-joint. A thumb is different from other fingers because it does not have second phalangeal bone and its metacarpophalangeal joint gives it significant opportunity of motion. Particular fingers are driven by system of tendons drawn by muscles, which are placed in forearm. The tendons are matched for every motion of the fingers. The first is for contraction and the second is for expansion of the finger. When the carpal bones are neglected the human hand has 24 + 6 degrees of freedom, 24 for hand and 6 for placement in 3D space, see Fig. 3a.



a) b)

Fig. 3. a) Kinematic configuration of human hand b) available joint motions

In the Fig.3b the two kinds of joints are shown. The first is revolute joint and the second is spherical joint. Our investigated humanoid hand will consists of these two joints.

3. Modeling of Humanoid Robot Hand in Matlab

The aim of this section is to create interface between user and model of humanoid robot hand. For this purposes will be used software Matlab using its toolboxes. By defining the basic parameters of hand the model should determines workspace of particular fingers, determines configuration variables for achievement of required points in the space by the end of finger, calculate constants of controller and simulate feedback control of position for particular fingers.

3.1. Blocks of Model

The model consists of together connected separate blocks. These blocks communicate to each other through the scripts. The architecture of designed model is shown in the Fig. 4.

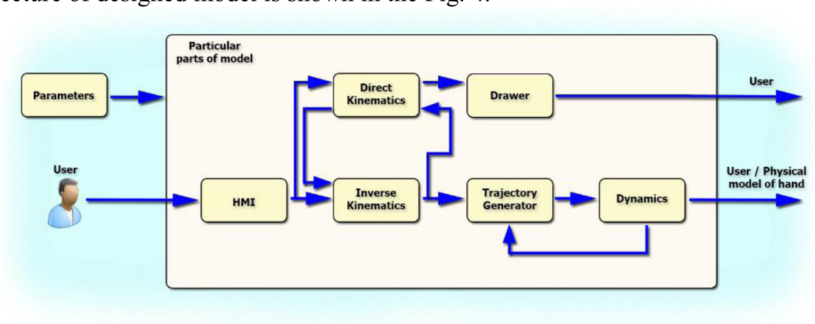


Fig. 4. Information flow between particular blocks

Model uses following blocks:

Block HMI – HMI (Human – Machine Interface) is hierarchical the highest block of whole model. This block coordinates functions of other block by running these block by means of defined order. HMI block ensures correct work of particular scripts of model. The user can enter input parameters like location of end of particular fingers.

Block Direct Kinematics – This block represents Matlab script consisting of equations for direct kinematic of particular fingers of hand. The input to this block are values of configuration variables given by user or output from block Inverse Kinematics. Next input to block Direct Kinematics are basic parameters of humanoid hand, which are necessary for calculation of kinematic equations. This block has two variations. Outputs of the first variant “A” are transformation matrices between particular phalanges. This variant can be used for verification purposes of results correctness of block Inverse Kinematics. The workspace of particular fingers is drawn using variant “A”. The second variant is “B”. Inputs to this block are only values of geometrical constants from the block Parameters. Outputs of this variant are transformation matrices between particular phalanges but only in symbolic form.

Block Parameters – Block contains all variables, which model uses during computing process. In other words, it contains weights, moments of inertia, geometrical dimensions of particular hand fingers. It contains also constants and particular tolerances, which are necessary during calculation process for example during PID controllers designing. Geometrical dimensions in this block were derived from the dimensions of human hand.

Block Drawer – The work of this block is to visualize actual configuration of particular fingers, which are determined by block Direct Kinematics variant “A”. Output of the block Drawer is not necessary for next calculations. User can verify by this way the correctness of computing process.

Block Inverse Kinematics – The role of this block is to determine required angles of configuration variables in order to reach required position of end of finger. Considering that each finger has at least four degrees of freedom the result of inverse kinematics is multivalued. Therefore the analytical solution cannot be determined. Block Inverse Kinematics uses numerical solution for calculation of configuration variables. [6, 9]

Block Trajectory – This block generates time behavior of particular joint rotation using initial values of joint positions, which are determined by block Inverse Kinematics. Output of this block is an input to block Dynamics.

Block Dynamics – Block Dynamics contains structures for humanoid hand motion control. Unlike other blocks which are in form of Matlab script, this block is designed in Matlab / Simulink, SimMechanics. Block Dynamics consists of two parts. The first part is set of controllers, which control particular fingers position. The second part is direct dynamic model of particular fingers. Block also contains system for verification of reached position of fingers.

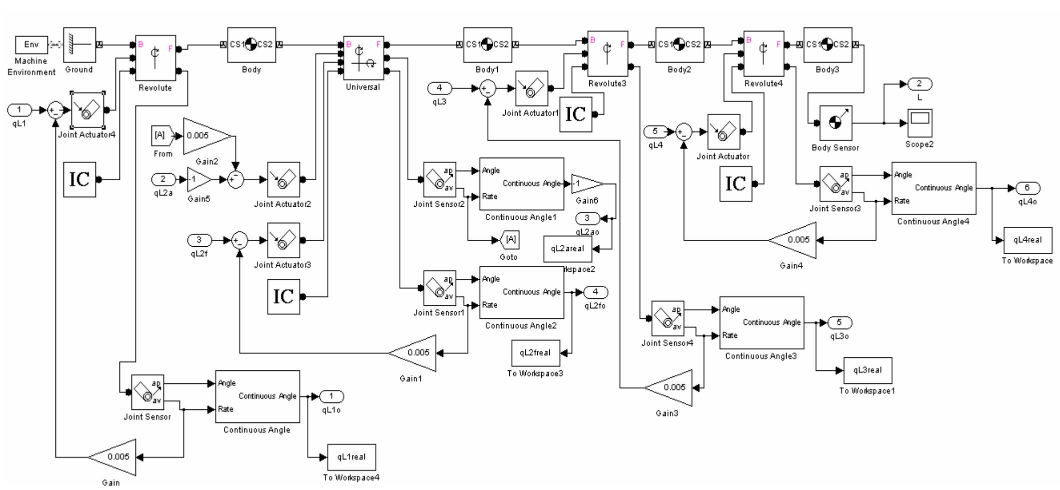


Fig. 5. Flow diagram of little finger using SimMechanics

3.2. Optimizing Algorithm for Inverse Kinematics

For optimization purposes of inverse kinematics the Matlab function *fmincon* is used. Function *fmincon* tries to find bounded minimum of scalar function of several variables starting with initial guess.

$$\min f(\mathbf{x}) for \left\{ \begin{array}{c} \mathbf{c}(\mathbf{x}) \leq 0 \\ \mathbf{ceq}(\mathbf{x}) = 0 \\ \mathbf{A}.\mathbf{x} \leq \mathbf{b} \\ \mathbf{Aeq}.\mathbf{x} = \mathbf{beq} \\ \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} \end{array} \right\} \tag{1}$$

where $\mathbf{x}$ – vector of independent variable, $\mathbf{lb}$ – vector of lower bounded values, $\mathbf{ub}$ – vector of upper bounded values, $\mathbf{c}(\mathbf{x})$, $\mathbf{ceq}(\mathbf{x})$ – functions x for conditions of equality and inequality, $\mathbf{A}$, $\mathbf{Aeq}$ – matrices of parameters for conditions of linear equality and inequality, $\mathbf{b}$, $\mathbf{beq}$ – vectors of right sides of conditions of linear equality and inequality, $f(\mathbf{x})$ – objective function of independent variables, from which the extreme is looking for.
Inverse Kinematics block in Fig. 6 is shown.

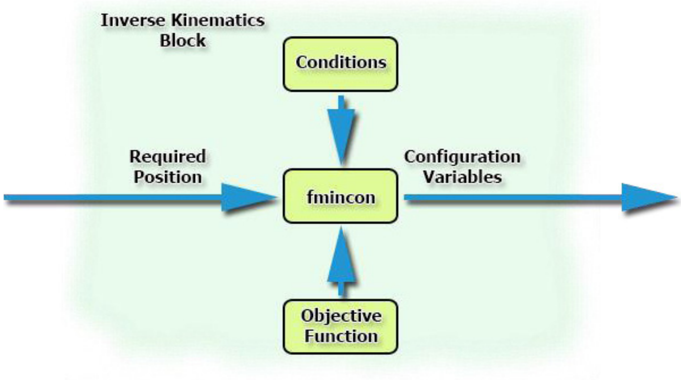


Fig. 6. Inverse Kinematics Block

Optimizing algorithm has to fulfil two conditions. The first is condition of function extreme and the second is fulfillment of bounded conditions. The second condition can be used for inverse kinematics solution. Bounded conditions are input to computing algorithm and they are written in following form:

$$\mathbf{ip}.\left(\left|\mathbf{p}_{iter}(\mathbf{q}_j) - \mathbf{p}_p\right| - tol\right) \leq 0 \tag{2}$$

where $\mathbf{p}_{\mathrm{iter}}$ – vector of positions of manipulator points for actual j-th iteration, $\mathbf{p}_{\mathrm{p}}$ – vector of required positions of manipulator points, *tol* – required accuracy of calculation, $\mathbf{q}_{\mathrm{j}}$ – vector of configuration variables for actual j-th iteration, $\mathbf{ip}$ – binary vector of values 0 and 1.

3.3. Local minimum problem

The method *fmincon* is gradient method. It tries to minimize deviation of actual position of end of finger for i-th iteration, from the area, which is bounded by conditions of solution. During computing the task of inverse kinematics can arise a situation, when the gradient of deviation does not decrease, while the conditions are not still fulfil. This problem concerns some points (red color), which are in specific area of finger workspace, see Fig. 7.

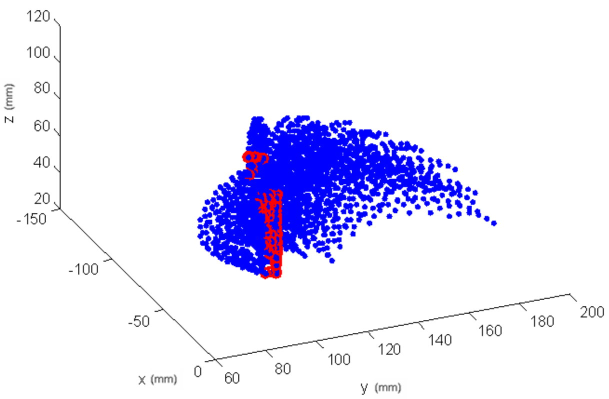


Fig. 7. Workspace of little finger with focus on area of errors

The workspace is divided by plane, in which the metacarpal link is parallel with axis *z*. This plane divides workspace to two half-areas. When searching of configuration for these points starts in half-area, from which it cannot be determined, the gradient of error decreases up to certain minimum. Then the computing algorithm cannot leave this local minimum of error gradient. The problem can be overcome by correcting algorithm. When too high error of solution is detected, the task will be calculating once more, while initial point of solution will be from second half-area.

3.4. Motion Control of Particular Fingers

For Motion control of particular fingers is used decentralized way of control. Decentralized way of control considers that all joints are independent to each other and influence of motion of other joints on one joint is assumed as disturbance. This is the simplest way of motion control often use to manipulator control. The constant of PID controllers are designed by Simulink Control Design. At first, PID tuner create linearized model of controlled system and then the constants of PID can be designed for chosen band width. By this way can be designed all PID constants for one finger. In matter of fact this method has several disadvantages. When the first controller is designed, then can be designed the second controller. Nevertheless, by designing the second controller, the first one will be out of tune. The result of mentioned approach results in oscillation of the finger, for example see Fig. 8.

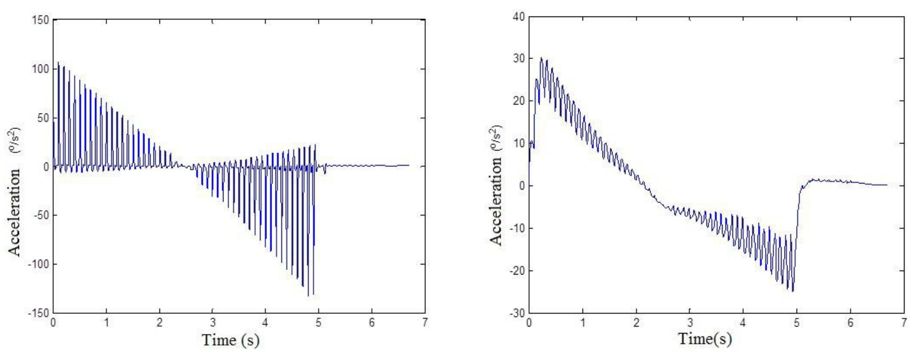


Fig. 8. a) Oscillation of finger using wrong PID constants b) Oscillation of finger using corrected PID constants

3.5. Calculation of PID Constants Using Optimizing Algorithm

From the Fig. 8a is obvious, that acceleration of finger is too high. For real application is this course of acceleration inappropriate. Using model-based approach by means of *Control Toolbox* is in this case ineffective. From this reason, the PID constants will be designed using *fmincon* function, mentioned above. The condition of solution is equality of the end of finger position (at the end of the simulation) with required position within the tolerance. The minimum step will be 0,001. The initial conditions are written in following table.

Tab. 2 Intinial conditions and ranges of PID constants

	P	I	D	N
Upper bound	1	1	0,1	6
Lower bound	0,01	0,01	-0,1	0
Initial condition	0,5	0,5	0	2

For correct function of optimizing algorithm were values for D constant hundred times enlarged and for N constant hundred times reduced. Values of optimized PID constants of particular joints are given in following tables.

Tab. 3 Constants of PID controllers for particular fingers

Joint	Constant	Little Finger	Ring Finger	Middle Finger	Forefinger
CM	P	0,0238	0,0623	0,0727	0,1823
	I	0,0193	0,0147	0,0222	0,1158
	D	7,68E-05	8,85E-05	-7,56E-05	-8,48E-05
	N	200,0008	290,8466	246,6096	448,337
MP (duction)	P	0,1009	0,0221	0,0281	0,0947
	I	0,0232	0,0138	0,0333	0,0581
	D	6,7E-05	-8,24E-05	8,15E-05	-1,99E-05
	N	199,9794	23,4336	200	359,8309
MP (flexion)	P	0,078	0,0525	0,067	0,05
	I	0,0335	0,0474	0,0569	0,1825
	D	-4,80E-05	8,14E-05	-3,43E-05	1,24E-05
	N	201,7216	405,0328	208,2643	396,5678
PIP	P	0,0945	0,044	0,0218	0,0707
	I	0,0653	0,0916	0,0134	0,0581
	D	-5,98E-05	-8,85E-05	5,39E-05	-2,77E-05
	N	202,0467	41,8132	107,834	385,5422
DIP	P	0,2302	0,0794	0,0312	0,0147
	I	0,0227	0,0141	0,018	0,1331
	D	-7,42E-05	9,04E-05	4,60E-05	4,25E-05
	N	202,3672	21,2992	446,2191	133,3575

Tab. 4 Constants of PID controllers for thumb

Joint	Constant	Thumb
CP (duction)	P	0,05
	I	0,05
	D	-2,03E-05
	N	200

MP (duction)	P	0,0576
	I	0,0577
	D	-7,96E-05
	N	199,5371
PIP	P	0,0554
	I	0,0556
	D	-7,96E-05
	N	198,8535
DIP	P	0,0534
	I	0,0497
	D	-7,96E-05
	N	198,575

4. Simulations and Results

The simulations are used for verification of above mentioned approaches. At first, drawn workspace of thumb and little finger are shown in the Fig. 9 – 10. Next, inverse kinematics solution is tested with using and without using of correcting algorithm. Subsequently, the determined constants of PID controllers are used for testing of hand dynamics. For testing was used Matlab r2013b.

4.1. Workspace of Thumb and Little Finger

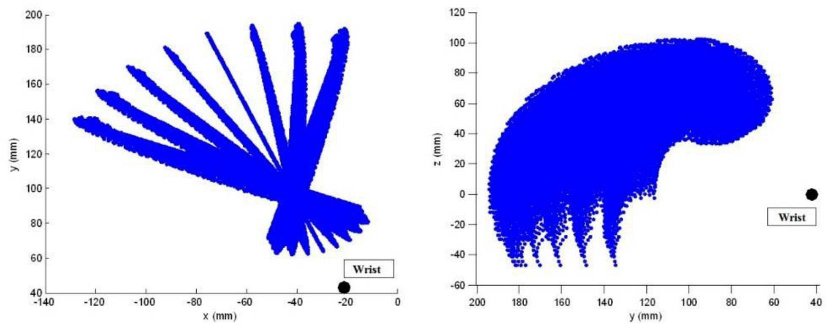


Fig. 9. Workspace of little finger

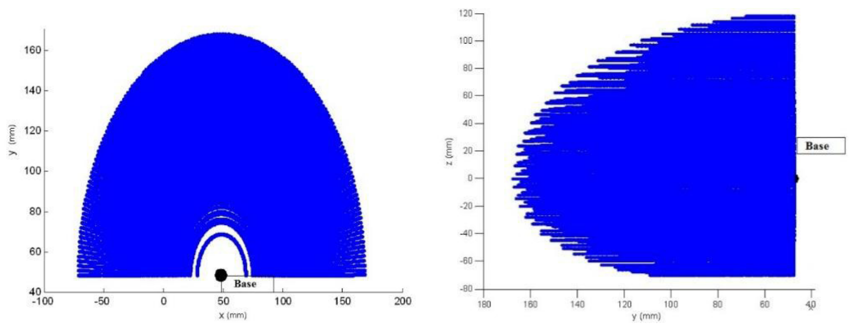


Fig. 10. Workspace of thumb

4.2. Test of Inverse Kinematic

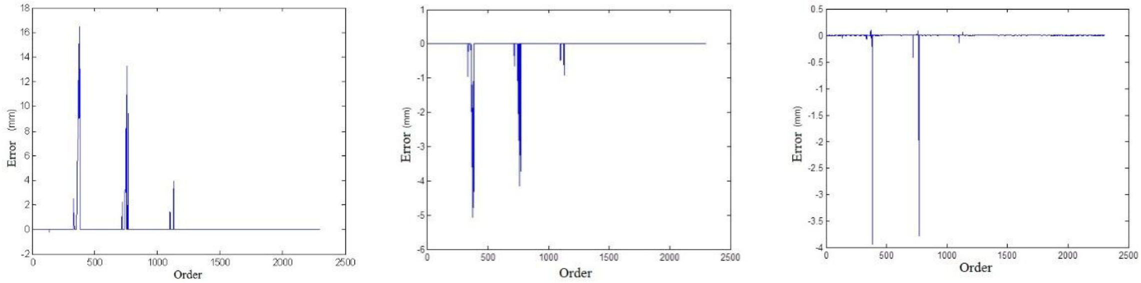


Fig. 11. Course of errors for x, y, z axis without using of corrective algorithm – little finger

In some point the errors are too high. From this reason is used correcting algorithm. Course of errors using this algorithm are shown in following figures.

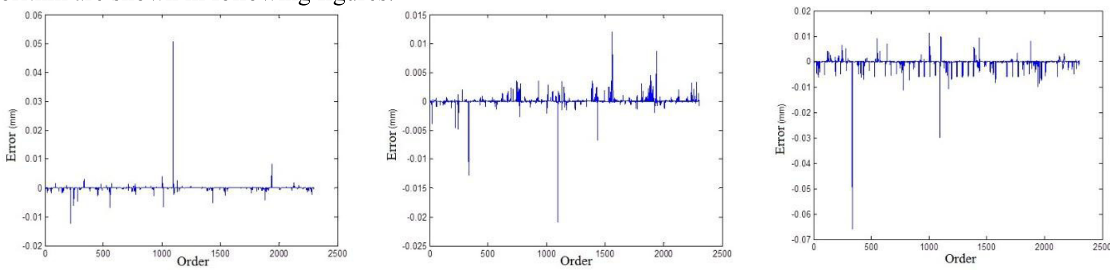


Fig. 12. Course of errors for x, y, z axis with using of corrective algorithm – little finger

Using correcting algorithm the size of errors rapidly decreases, almost thousand times.
Now, the inverse kinematics will be tested by demonstrating on simple example. The task is to push a button by middle finger.

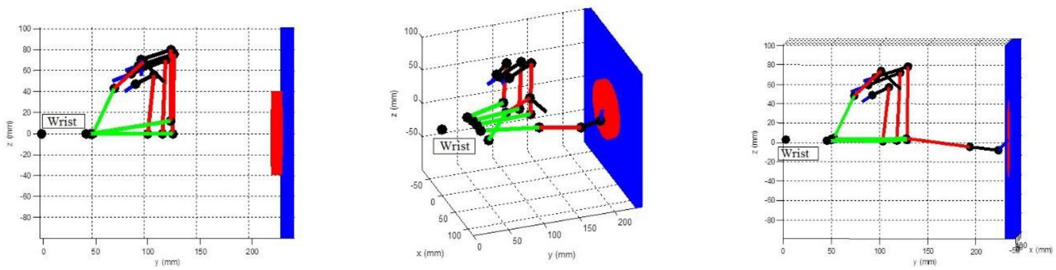


Fig. 13. Pushing the button by middle finger – example of inverse kinematics

4.3. Test of Dynamics

For test of dynamics the PID controllers are used. The constants of PID are determined in section *3.5*. Time courses of particular joints in the Fig. 14 are shown.

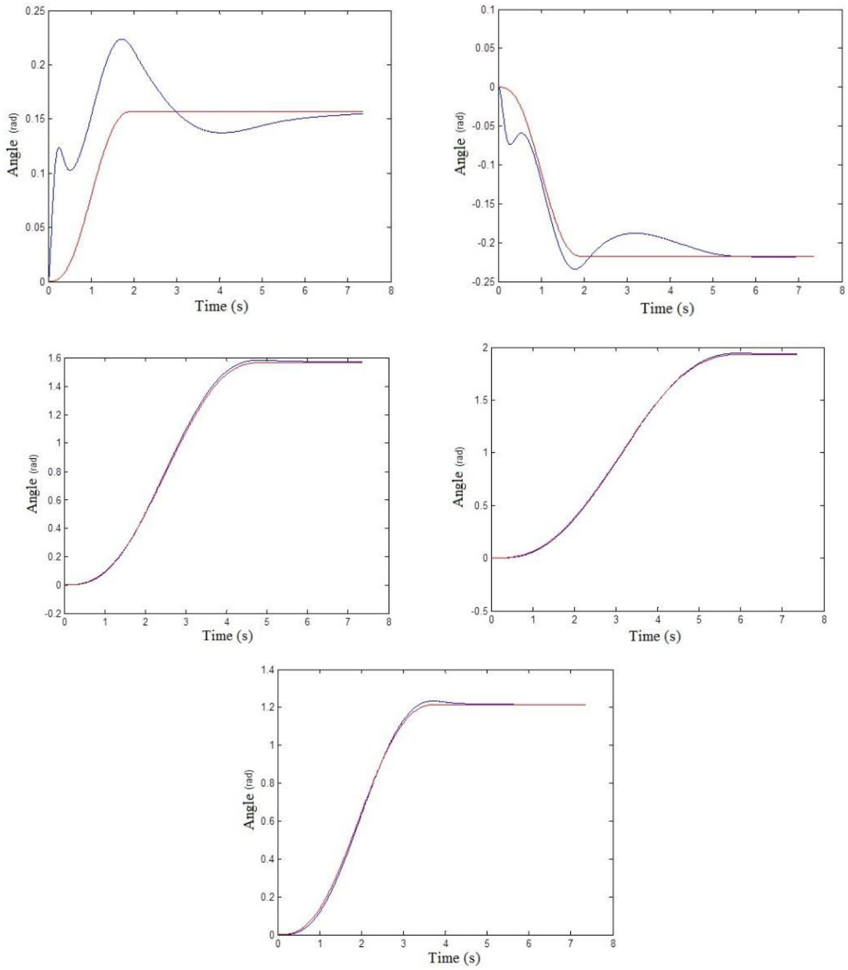


Fig. 14. CM joint, MP (duction) joint, MP (flexion) joint, PIP joint, DIP joint

5. Conclusion

The paper presents approach to modeling of motion of humanoid robot hand in Matlab. Model consists of direct and inverse kinematics, trajectory generator, drawer, and dynamics block. For inverse kinematics solution and PID constants determination the Matlab method *fmincon* is used. Difficulties and problems arising during computing was corrected by correcting algorithm. The paper introduced the model which can be used for control of physical model of humanoid hand by some measurement card, which communicates with matlab.

Acknowledgements

The authors would like to thank to Slovak Grant Agency – project VEGA 1/1205/12 "Numerical modelling of mechatronic systems". This contribution is also result of the project APVV-0091-11 “Using of methods of experimental and numerical modeling for increasing of competitiveness and innovation of mechanical and mechatronical systems” and KEGA project no. 048TUKE-4/2014 “Increasing of knowledge base of students in area of application of embedded systems in mechatronic systems”.

References

[1] J. Hirt, K. Berns, K. Mianowski, Designing Arms and Hands for the Humanoid Robot ROMAN, in: Advanced Materials Research (2012), pp. 1233 – 1237.

[2] S. Kagami, M. Mochimaru, Y. Ehara, N. Miyata, K. Nishiwaki, T. Kanade, and H. Inoue, Measurement and comparison of human and humanoid walking, in Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, Vol. 2, 2003, pp. 918 – 922.

[3] J. de Lope, T. Zarraonandia, R. González-Careaga, D. Maravall, Solving the Inverse Kinematics in Humanoid Robots: A Neural Approach, Lecture Notes in Computer Science (2003), Vol. 2687, pp. 177 – 184.

[4] C. L. Taylor, R. J. Schwarz, The Anatomy and Mechanics of the Human Hand, Artificial Limbs (1955), Vol. 2, No. 2, pp. 22 – 35

[5] R. Spielmann, F. Duchoň, J. Kostroš, T. Fico, R. Balog, Sensor Module for Mobile Robot, in: American Journal of Mechanical Engineering (2013), Vol. 1, No. 7, 378-383

[6] S. R. Buss, Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods, IEEE Journal of Robotics and Automation (2004), pp. 681 – 685.

[7] K. Trnka, P. Božek, Optimal Motion Planning of Spot Welding Robot Applications. In: Applied Mechanics and Materials, ICMMME 2012, ISSN 1660-9336, pp. 589-593.

[8] F. Duchoň, P. Hubinský, J. Hanzel, A. Babinec, M. Tölgyessy, Intelligent vehicles as the robotic applications, in: Procedia Engineering 48 (2012), Vol. 48, pp. 105-114

[9] A. Schmitz, M. Maggiali, L. Natale, G. Metta, Touch Sensor for Humanoid Hands, IEEE Symposium on Robot and Human Interactive Communication, 2010, pp. 691 – 697.

[10] D. Koniar, L. Hargaš, M. Hrianka, V. Bobek, P. Drgoňa, P. Fibich, Kinematics analysis of biomechanical systems using image analysis, Metalurgija (Metallurgy), Vol. 49 (2/2010), ISSN 1334-2576.